

**NPS ARCHIVE**  
**1969**  
**READ, D.**

AN ITERATIVE PROCESS TO SOLVE  
THE GRAPH-COLORING PROBLEM

by

Dennis Spencer Read



# United States Naval Postgraduate School



## THESIS

AN ITERATIVE PROCESS  
TO  
SOLVE THE GRAPH-COLORING PROBLEM

by

Dennis Spencer Read

T132664

October 1969

*This document has been approved for public re-  
lease and sale; its distribution is unlimited.*



An Iterative Process  
to  
Solve the Graph-Coloring Problem

by

Dennis Spencer Read  
Lieutenant Commander, United States Navy  
B.S., United States Naval Academy, 1959

Submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL  
October 1969

NPS ARCHIVE

~~42-61 K25505~~ c.1

1969

RENO, D.

ABSTRACT

The intent of this paper is to describe a method of coloring a map and to present an algorithm for the solution of this problem. A computer program was developed to provide solutions to the problem of coloring a map which consists of a finite number of areas. This algorithm may also be applied to problems other than map-coloring.

TABLE OF CONTENTS

I. INTRODUCTION ----- 9

II. THE COLORING PROCESS ----- 11

    A. THE ALGORITHM ----- 11

    B. EXAMPLE ----- 14

III. CONCLUSION ----- 18

APPENDIX A. SAMPLE PROBLEM SOLUTION ----- 19

APPENDIX B. PROGRAM FLOW CHART ----- 21

COMPUTER PROGRAM ----- 26

BIBLIOGRAPHY ----- 39

INITIAL DISTRIBUTION LIST ----- 40

FORM DD 1473 ----- 41





## LIST OF ILLUSTRATIONS

FIGURE	TITLE	PAGE
1	Possible path to a solution to the map-coloring problem -----	13
2	Example problem -----	14



# TABLE OF SYMBOLS

SYMBOL	MEANING
$M$	Number of equations used to describe the border constraints for a map.
$N$	Number of areas to be colored
$x_i$	Variable representing the color of area $i$
$A_{M \times N}$	The $M \times N$ matrix defining regions having common borders
$\bar{a}_j$	The $j^{\text{th}}$ column of $A_{M \times N}$
$\underline{a}_i$	The $i^{\text{th}}$ row of $A_{M \times N}$
$A'_{M \times N}$	$A_{M \times N} + [\bar{a}_j, \dots, \bar{a}_j]_{1 \times N}$
$\beta_j$	The number of zero elements of each column $\bar{a}_j$ that are not associated with the constraint equations
$\alpha_j$	The number of zero elements of each column $\bar{a}_j$



## I. INTRODUCTION

When coloring a geographical map, it is customary to use different colors for adjacent regions - where adjacent means a common finite boundary. It has been conjectured, but not proved, that "For any subdivision of the plane into non-overlapping regions, it is always possible to mark the regions with one of the numbers 0, 1, 2, 3, in such a way that no two adjacent regions receive the same number."<sup>1</sup>

The statement that a map can be colored with four colors is accredited to Moebius, who first proposed it in 1840. This conjecture has never been proven; however, a satisfactory counter-example has not been demonstrated either. A proof for five-coloring a map may be found in [Courant and Robbins 1941] and [Oystein 1967].

This thesis will use the four-coloring conjecture without proof and present an algorithm that utilizes an iterative routine to determine what color each region should be. A similar iterative technique for general integer programming problems is demonstrated in [Greenberg April 1969] and [Greenberg May 1969], which present the proofs. We are interested in showing whether the iterative technique can be used for large-scale problems such as the map-coloring problem.

The problem formulation suggested by Gomory [Dantzig 1960] is used. Let the map regions be denoted by  $i = 1, 2, \dots, N$  and let  $x_i$  be an integer valued variable such that

$$0 \leq x_i \leq 3 \quad (1)$$

---

<sup>1</sup>Courant, Richard and Robbins, Herbert; What is Mathematics?, p. 247, Oxford University Press, 1941.

where the four values  $x_i = 0, 1, 2, \text{ or } 3$  correspond to four different colors. Since any two adjacent areas must have different colors, the constraint may be written in an either/or form

$$\text{either } x_i - x_j \geq 1 \quad \text{or} \quad x_j - x_i \geq 1 \quad (2)$$

if  $i$  and  $j$  have a common border. Equation (2) may be rewritten as:

$$\begin{aligned} x_i - x_j &\geq 1 - 4\delta_{ij} & (\delta_{ij} = 0, 1) \\ x_j - x_i &\geq -3 + 4\delta_{ij} \end{aligned} \quad (3)$$

Instead of using (2) and (3) we can simplify the problem by using

$$x_i - x_j \neq 0 \quad (4)$$

where  $i$  and  $j$  have a common border.

This formulation can be written in matrix form

$$Ax \neq 0 \quad (5)$$

where the elements of  $A_{M \times N}$ ,  $a_{ij}$ , are either 0, -1, or 1. We can now enumerate all values of the left side of (5). The column with the least number of zeroes is now selected from  $A_{M \times N}$ ,  $\bar{a}_j$ ,<sup>2</sup> and the value of the corresponding  $x_j$  is set equal to one. The column  $\bar{a}_j$  is then added column by column to the matrix  $A_{M \times N}$  to form a new matrix. The process continues in the same way by searching the new matrix for the column with the least number of zeros. In each iteration the corresponding  $x_j$  is increased by one. The column values represent the value of the left side of (5). A solution occurs when a column is achieved that has no zeros.

---

<sup>2</sup>The speed with which the algorithm reaches a solution is increased by modifying the method of choosing  $\bar{a}_j$ . This modification is explained in Section II. A. 4.



The algorithm presented is based upon the four-color conjecture, but is not restricted to the use of only four colors, if more colors are desired.

## II. THE COLORING PROCESS

### A. THE ALGORITHM

1. Let  $x_i = 0$  for  $i = 1, \dots, N$ .
2. For every 2 regions of the map with a common border it is required that  $x_i - x_j \neq 0$ ,  $i, j, = 1, \dots, N$ ;  $i \neq j$ .
3. Form the  $M \times N$  matrix ( $A_{M \times N}$ ) defined in step 2, where  $M$  is the number of equations required to describe the border constraints and  $N$  is the number of areas to be colored. The elements of  $A_{M \times N}$ ,  $a_{ij}$ , are either -1, 0, or 1. Note:
  - a. For each row,  $\underline{a}_i$ ,  $i = 1, \dots, M$ , there are only two non zero elements, -1, and 1, corresponding to the constraints in step 2.
  - b. For column  $\bar{a}_j$ ,  $j = 1, \dots, N$ , has at least one non zero element and fewer than  $M$  non zero elements.
4. Let  $\alpha_j$ ,  $j = 1, \dots, N$ , be the number of zero elements of each  $\bar{a}_j$ . Let  $\beta_j$ ,  $j = 1, \dots, N$ , be the number of zero elements of each  $\bar{a}_j$  which appear in those positions where  $a_{ij} = 0$ . Note:  $\beta_j \leq \alpha_j$ ,  $j = 1, \dots, N$ . Find the column of  $A_{M \times N}$  with the smallest  $\beta_j$ . Let  $j^{**}$  be the smallest  $j$  such that  $\beta_{j^{**}} = \min_j \{\beta_j\}$ . Let  $j^* = j^{**}$ .
5. Let  $x'_{j^*} = x_{j^*} + 1$ . Check all the constraints in which  $x'_{j^*}$  appears to insure that the constraint equations (4) are satisfied. If the constraint equations (4) are not satisfied, continue increasing  $x_{j^*}$  in increments of one until either (4) is satisfied or  $x'_{j^*} = 3$ . If  $x'_{j^*} = 3$  and (4) is not satisfied, set  $x'_{j^*} = 0$  and let  $j^*$  be the

smallest  $j > j^{**}$  such that  $\beta_{j^*} = \min_j \{\beta_j\}$  and repeat step 5. If no such  $j^*$  exists, then find the smallest  $\alpha_j$  and proceed as with the  $\beta_j$ 's. If no  $\alpha_j$  exists such that  $x_{j^*}$  and  $x_i$  satisfy (4) for all  $i \neq j^*$ , go to step 9.

6. Determine the elements of column  $\bar{a}_{j^*}$  by substituting the presently defined values of  $x_i$ ,  $i = 1, \dots, N$ , into the constraint equations (4). Add the column  $\bar{a}_{j^*}$ ,  $j = 1, \dots, N$ , to each column of  $A_{M \times N}$ . Let  $A'_{M \times N} = A_{M \times N} + [\bar{a}_{j^*}, \dots, \bar{a}_{j^*}]$ .

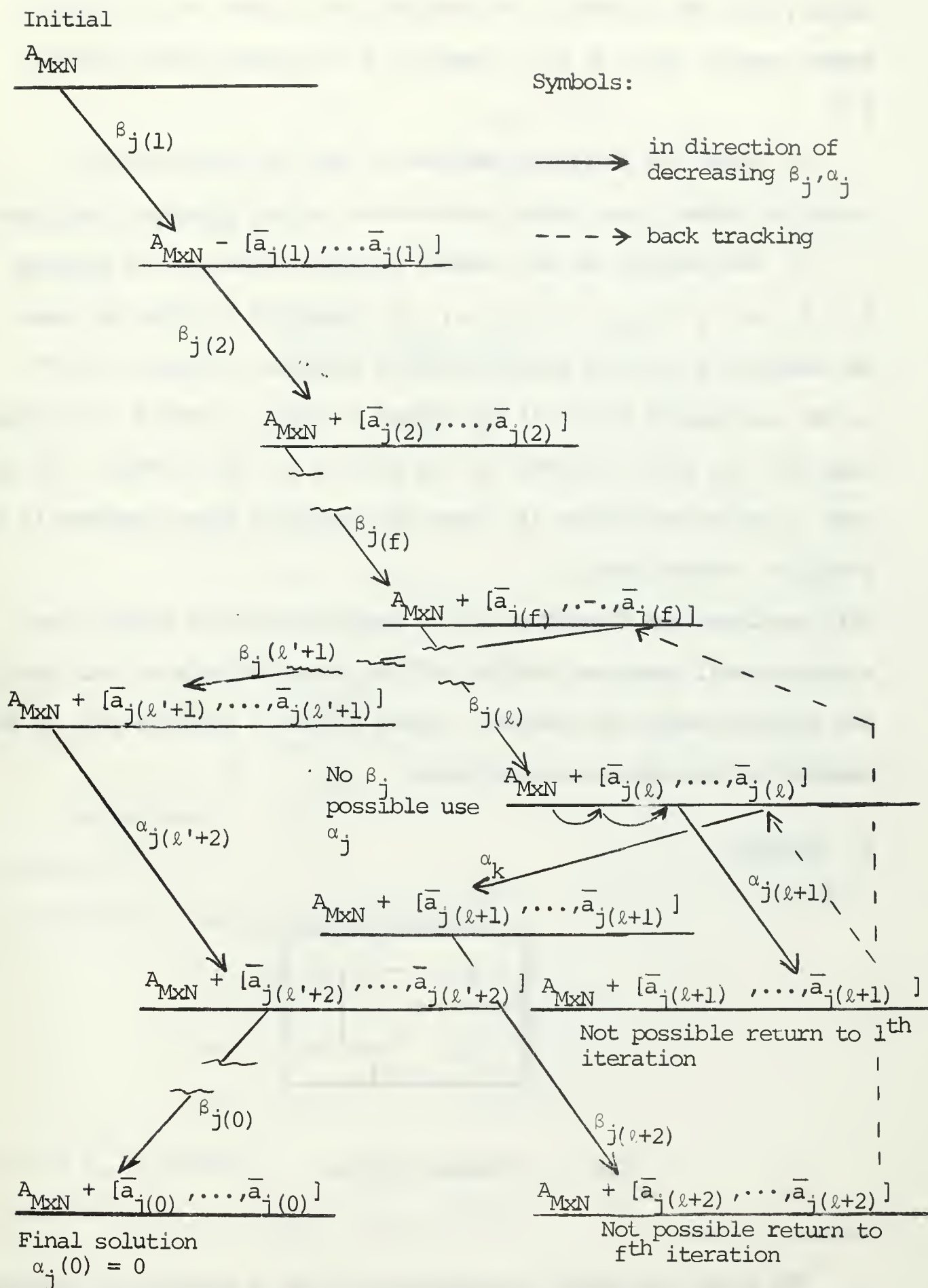
7. Let  $\alpha_j$ ,  $j = 1, \dots, N$ , be the number of zero elements of each  $\bar{a}'_j$ . Let  $\beta_j$ ,  $j = 1, \dots, N$ , be the number of zero elements of each  $\bar{a}'_j$  which appear in those positions where  $a_{ij} = 0$ . Let  $j^{**}$  be the smallest  $j$  such that  $\beta_j = \min_j \{\beta_j\}$ . Let  $j^* = j^{**}$ .

8. Continue steps 5 through 8 until  $\alpha_j = 0$ , in which case a solution has been reached. Note: If  $x_j = 3$  at any point during steps 5 through 8, disregard the corresponding column  $\bar{a}_j$  in any future iteration, since the corresponding  $x_j$  is at its maximum possible value.

9. Let the column  $\bar{a}_{j^*}$  be the one used to generate the latest  $A'_{M \times N}$  matrix. For each zero element of  $\bar{a}_{j^*}$  there are 2 regions,  $g$  and  $h$ , with a common border where  $x_g - x_h = 0$ , i.e., a violation of (4). Let  $F$  be the set of all such regions  $g$  and  $h$ . Denote by the subscript  $f$  the element of  $F$ .

10. For each  $f$ , increment the  $x_f$ 's such that  $x_f = x_f + 1$ . Check all of the constraints in which  $x_f$  appears to see if the constraint equations (4) are satisfied. If for any  $f$ , an element of  $F$ , the constraint equations (4) are satisfied set  $f = j^*$  and go to step 6. If (4) is not satisfied for any  $i$ , not an element of  $F$ , where regions  $i$  and  $f$  have a common





Possible Path to a solution to the map-coloring problem

Figure 1

border, note the iteration that defined the present value of  $x_i$ .

Repeat step 10 until  $x_f = 3$ . Reset  $x_f = 0$ . Repeat step 10 for all  $F$ .<sup>3</sup>

11. Order the iterations defined in step 10 in decreasing iteration number. Let these iterations be called possible iterations.

12. Reconstruct the most recent possible iteration and examine  $\beta_j = \beta_{j^*}$  or  $\alpha_j = \alpha_{j^*}$ ,  $j = j^* + 1, \dots, N$ , depending on which was used to determine  $x_{j^*}$  in the iteration being examined. Choose a new  $j^*$  in the same manner as the  $j^*$  was chosen in step 5. When a  $j^*$  is found such that  $x_{j^*}$  and  $x_i$  satisfy (4) for all  $i \neq j^*$ , go to step 6. If no such  $j^*$  exists satisfying (4), then the iteration being examined is not possible. Repeat step 12.

This completes the algorithm. If no possible solution exists, the algorithm will continue "backing up" the generated matrix chain until the original matrix is reached. Figure 1 shows a possible path to the solution of the map-coloring problem.

#### B. EXAMPLE

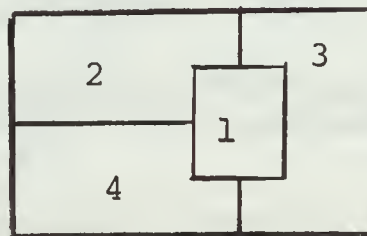


Fig. 2. Example Problem

---

<sup>3</sup>The speed with which the algorithm reaches a solution is increased by starting the backtracking procedure when no variable associated with the minimum  $\beta_j$  or  $\alpha_j$  can be set at any integer value between 1 and 3.

where

$$\begin{array}{ll} x_1 - x_2 \neq 0 & x_2 - x_3 \neq 0 \\ x_1 - x_3 \neq 0 & x_2 - x_4 \neq 0 \\ x_1 - x_4 \neq 0 & x_3 - x_4 \neq 0 \end{array}$$

set  $x_i = 0$  for all  $i$

we write

	$x_1$	$x_2$	$x_3$	$x_4$
	1	-1	0	0
	1	0	-1	0
	1	0	0	-1
	0	1	-1	0
	0	1	0	-1
	0	0	1	-1
$\alpha_j$	3	3	3	3
$\beta_j$	3	3	3	3

column used \*

then  $x_1 = 1$

the column to add  $(\bar{a}_{j*})$  is defined by:

$$\begin{array}{ll} x_1 - x_2 = 1 & x_2 - x_3 = 0 \\ x_1 - x_3 = 1 & x_2 - x_4 = 0 \\ x_1 - x_4 = 1 & x_3 - x_4 = 0 \end{array}$$

where  $\bar{a}_{j*}^T = [111000]$

then adding  $[\bar{a}_{j*}, \dots, \bar{a}_{j*}] + A_{M \times N}$  we write

	$x_1$	$x_2$	$x_3$	$x_4$
	2	0	1	1
	2	1	0	1
	2	1	1	0
	0	1	-1	0
	0	1	0	-1
	0	0	1	-1
$\alpha_j$	3	2	2	2
$\beta_j$	3	1	1	1

column used \*

then  $x_2 = 2$

the column to add ( $\bar{a}_{j*}$ ) is defined by:

$$\begin{array}{ll} x_1 - x_2 = -1 & x_2 - x_3 = 2 \\ x_1 - x_3 = 1 & x_2 - x_4 = 2 \\ x_1 - x_4 = 1 & x_3 - x_4 = 0 \end{array}$$

where  $\bar{a}_{j*}^T = [-111220]$

then adding  $[\bar{a}_{j*}, \dots, \bar{a}_{j*}] + A_{M \times N}$  we write

	$x_1$	$x_1$	$x_3$	$x_4$
	0	-2	-1	-1
	2	1	0	1
	2	1	1	0
	2	3	1	2
	2	3	2	1
	0	0	1	-1
$\alpha_j$	2	1	1	1
$\beta_j$	1	1	0	0

column used \*

then  $x_3 = 3$

the column to add  $(\bar{a}_{j*})$  is defined by:

$$\begin{array}{ll} x_1 - x_2 = -1 & x_2 - x_3 = -1 \\ x_1 - x_3 = -2 & x_2 - x_4 = 2 \\ x_1 - x_4 = 1 & x_3 - x_4 = 3 \end{array}$$

where  $\bar{a}_{j*}^T = [-1 \ -2 \ 1 \ -1 \ 2 \ 3]$

then adding  $[\bar{a}_{j*}, \dots, \bar{a}_{j*}] + A_{M \times N}$  we write

$$\begin{array}{cccc} & x_1 & x_2 & x_3 & x_4 \\ & \hline & 0 & -2 & -1 & \\ -1 & & -2 & -3 & \\ 2 & & 1 & 1 & \\ -1 & & 0 & -2 & \\ 2 & & 3 & 2 & \\ 3 & & 3 & 4 & \\ \hline \alpha_j & 1 & 1 & 0 & \\ & & & * & \end{array}$$

there is no need to calculate column  $\bar{a}'_4$  because there are no zeros in column  $\bar{a}'_3$  and a solution has been reached. The solution is:

Area number	Color number
1	1
2	2
3	3
4	0

### III. CONCLUSION

The algorithm described will solve the map-coloring problem. Although only four colors were used in the example in Section II-B, the algorithm can be adapted to solve a problem for any feasible number of colors. Several representative map-coloring problems were solved using the enclosed computer program. Appendix A shows a solution to four-coloring the continental United States. In the representative problems solved, both four and five colors were used.

If the solution is not feasible, e.g., the use of four colors when five colors are required, as in the split-state problem, the algorithm may consume several hours of computer time before showing that no solution exists. The computer time is consumed in back-tracking the solution path and in investigating every possible solution path for both the  $\alpha$ 's and the  $\beta$ 's.



## APPENDIX A

### SAMPLE PROBLEM SOLUTION

The solution for four-coloring the continental United States of America, including boundaries and major water bodies, is as follows:

STATE	COLOR
Maine	0
New Hampshire	3
Vermont	0
Massachusetts	2
Rhode Island	3
Connecticut	0
New York	3
New Jersey	0
Pennsylvania	2
Delaware	3
Maryland	0
Virginia	2
North Carolina	3
South Carolina	0
Georgia	2
Florida	0
Alabama	3
Tennessee	1
West Virginia	1
Ohio	0
Indiana	2
Kentucky	3
Illinois	0
Wisconsin	2
Michigan	3
Minnesota	0
Iowa	3
Missouri	2
Arkansas	3
Mississippi	2
Louisiana	0
Texas	2
Oklahoma	0
Kansas	3
Nebraska	0
South Dakota	1
North Dakota	3
Montana	0
Wyoming	2

Colorado	1
New Mexico	3
Arizona	1
Utah	3
Idaho	1
Washington	0
Oregon	2
Nevada	0
California	3

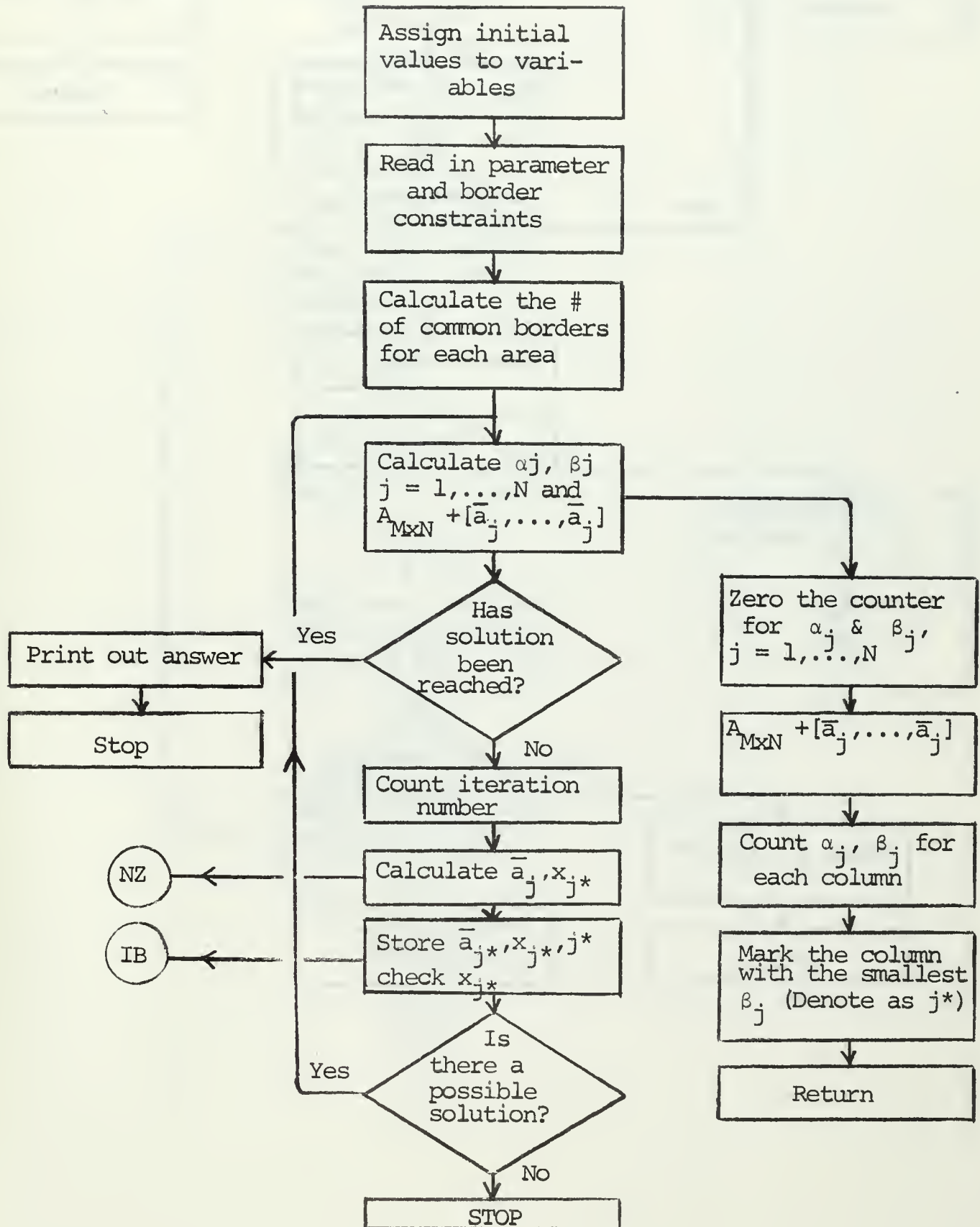
#### BORDER AREAS

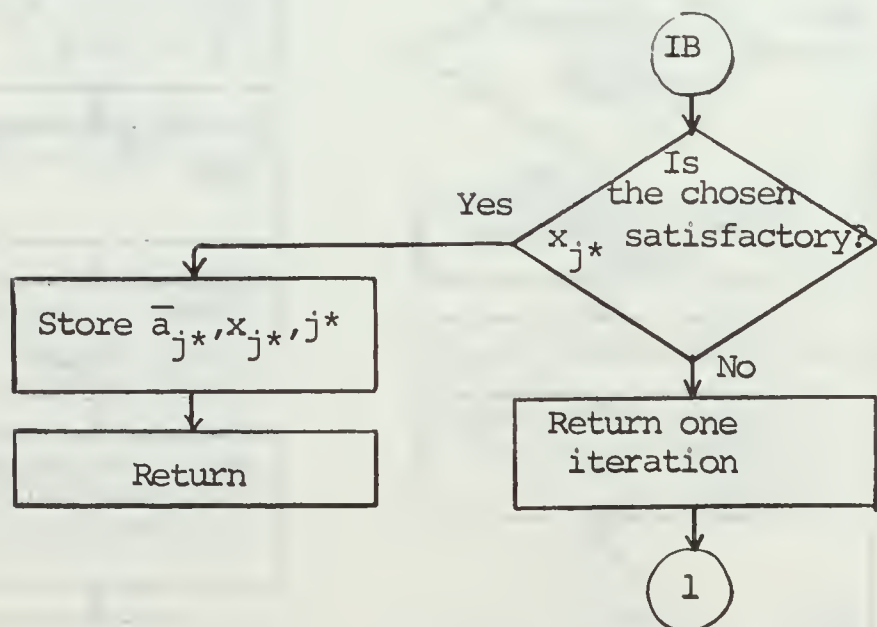
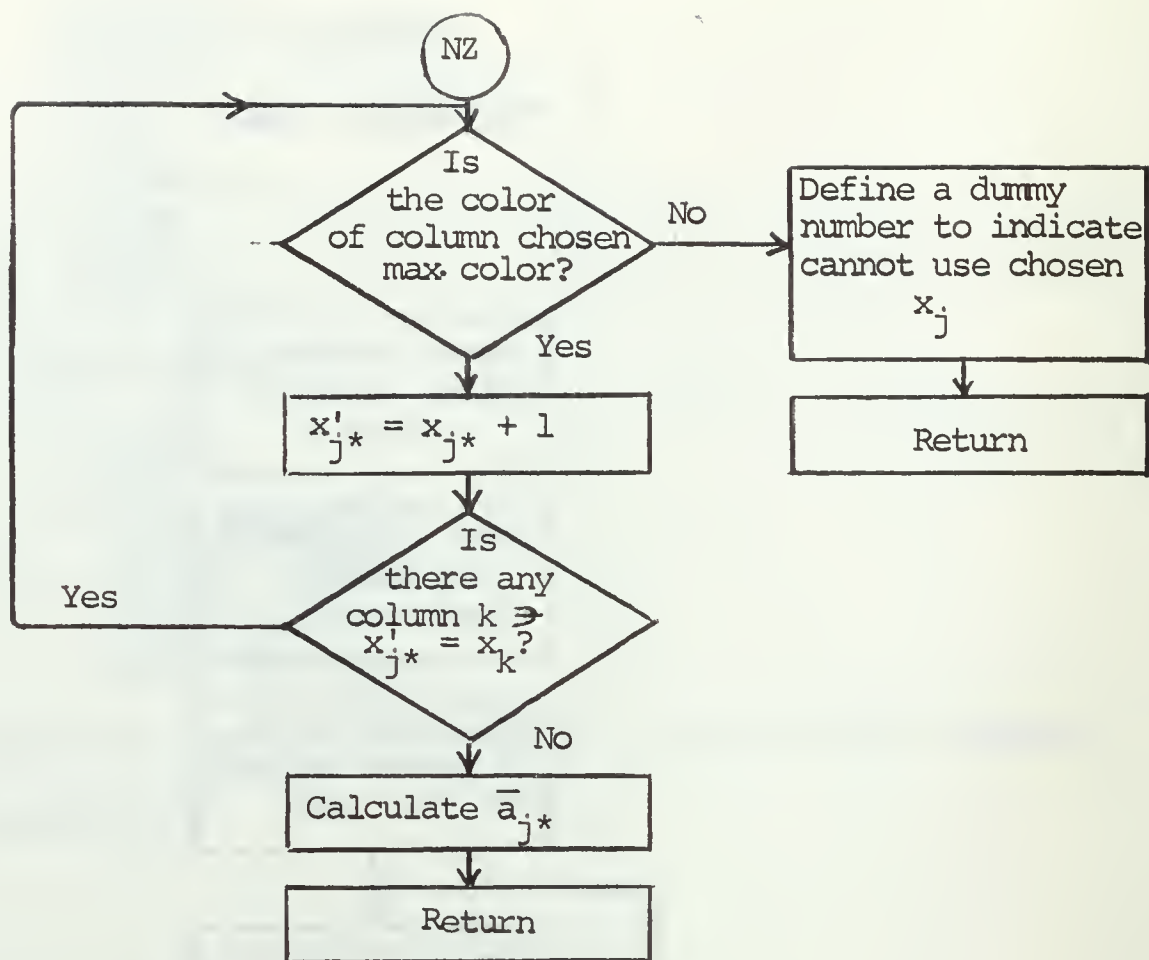
Canada	2
Mexico	0
Water - including:	1
Atlantic Ocean	
Pacific Ocean	
Great Lakes	
Great Salt Lake	

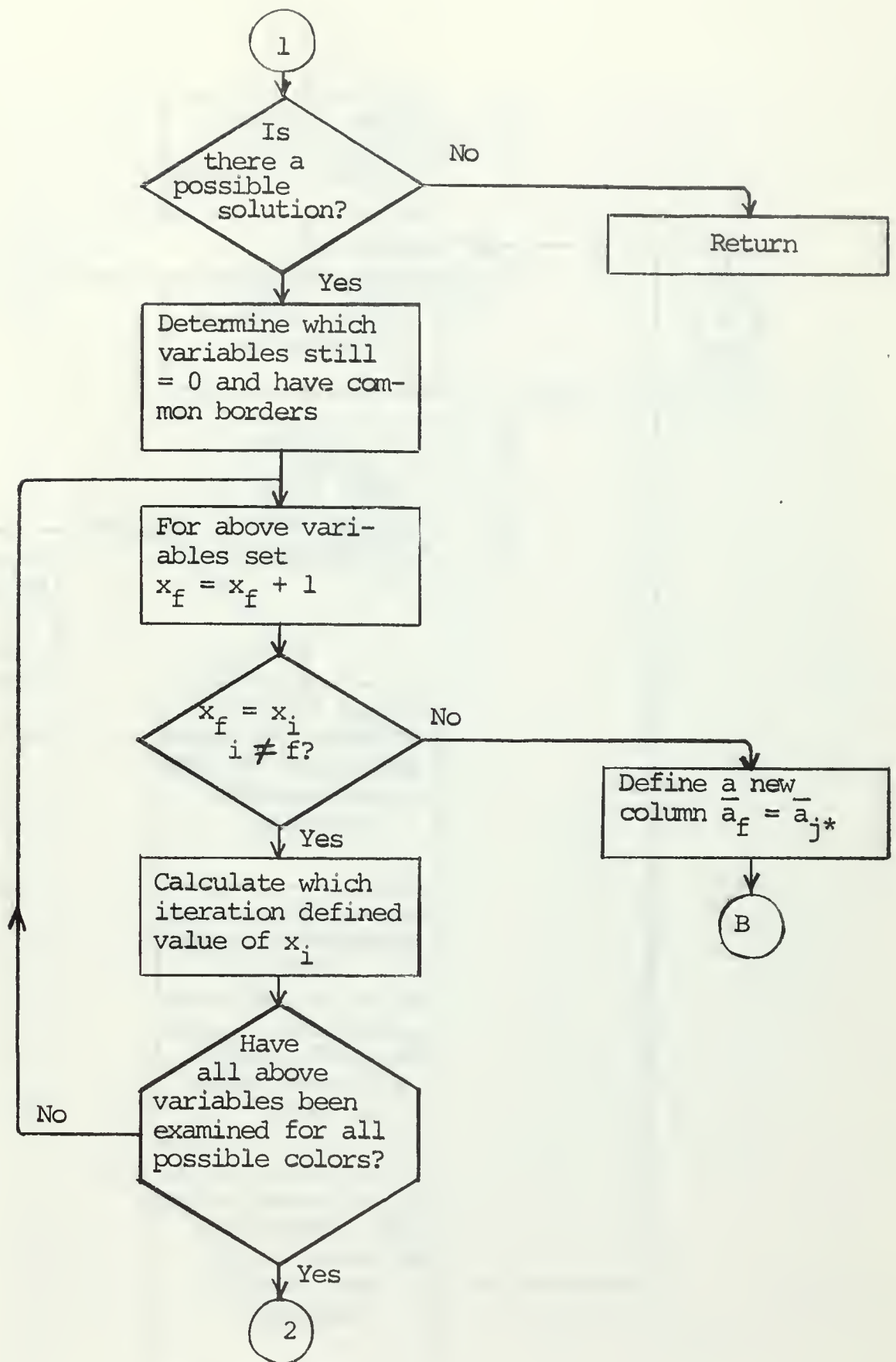
The above problem was solved in 97.2 seconds on the IBM 360/67 computer and used 81 iterations.

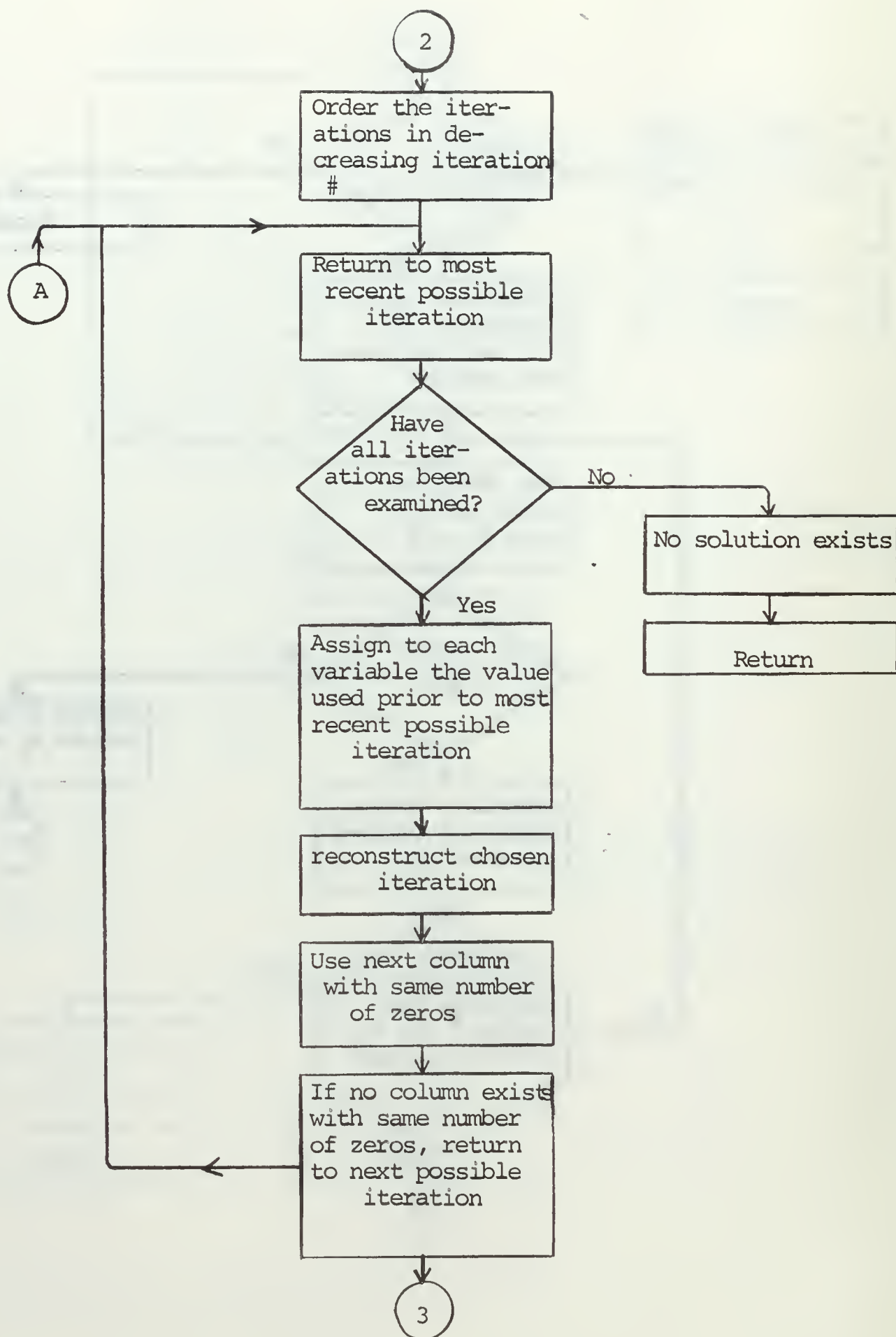


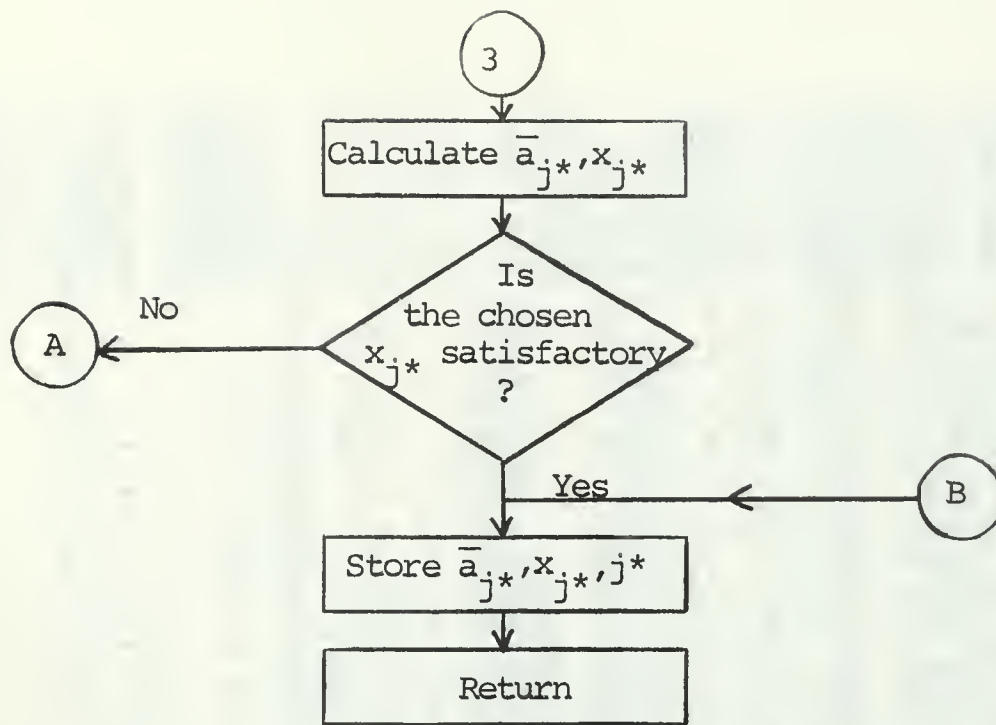
APPENDIX B  
PROGRAM FLOW CHART















```

110 MV(I)=0
115 DO 115 I=1,N
115 IADD(I)=0
117 DO 117 I=1,N
117 DO 117 J=1,NN
117 ITAB(I,J)=0
C READ IN THE INITIAL VALUES DEFINING BORDER AREAS
119 READ(5,120) I,J,K,ITAB(I,J),ITAB(I,K),IDUM
120 FORMAT(6I5)
C IF(IDUM.EQ.0) GO TO 119
C DETERMINE THE NUMBER OF BORDERING AREAS FOR EACH STATE
DO 121 J=1,NN
DO 121 I=1,N
IJ=ITAB(I,J)
121 KOLM(J)=KOLM(J)+IABS(IJ)
C DETERMINE THE COLUMN TO USE FOR THE NEXT ITERATION AND MARK IT
125 CALL NMTX(NN,N,ITAB,IADD,MAT,II,DUM,NZ,ITSV,NZCON)
C IF SOLUTION HAS BEEN REACHED, PRINT OUT THE ANSWER
C IF(DUM.GT.0.0) GO TO 130
C INCREASE COUNTER BY ONE
KAT=KAT+1
C DETERMINE THE NEW VALUE OF THE VARIABLE, INDICATE IF SOLUTION IS
C INFEASIBLE SO THAT A NEW PATH CAN BE TAKEN, AND DEFINE THE NEW COLUMN
C TO BE ADDED TO THE INITIAL TABLEAU
140 CALL NZERO(NN,II,ITAB,ITSV,MV,E,KLR,IADD,BA,KOLM)
C IF(BA.NE.0.0) GO TO 140
C SAVE THE COLUMN ADDED TO THE INITIAL TABLEAU/VARIABLE USED. IF SOLUTION
C FEASIBLE, GO BACK TO FIRST FEASIBLE TABLEAU AND PICK A NEW VARIABLE TO
CALL IBACK(NN,IADD,KNT,II,NZ,ITAB,ITSV,DUM,MV,MAT,E,KLR,KOLM,
INZCON,NP3,NNX3)
IF(DUM.GT.0.0) GO TO 130
GO TO 125
C WRITE OUT COLORS FOR EACH AREA.
130 DO 135 I=1,NN
WRITE(6,150) I,MV(I)
150 FORMAT(10X,'FOR AREA NUMBER USE COLOR NUMBER',T27,I3,T48,I3)
135 CONTINUE
STOP
END

SUBROUTINE NMTX(NN,N,IM,IT,MT,II,D,NZ,ISV,NZCON)
SUBPROGRAM TO DO THE FOLLOWING:
(1) ADD COLUMN WITH LEAST NUMBER OF ZEROS TO INITIAL MATRIX
(2) IF ANY COLUMN HAS NO ZEROS STOP THE ITERATION
SYMBOLS ARE:
MT=MAT; IM=ITAB,D=DUM, IT=IADD, ISV=ITSV

```

MAIN0058  
 MAIN0060  
 MAIN0080  
 MAIN0090  
 MAIN0095  
 MAIN0096  
 MAIN0097  
 MAIN0100  
 MAIN0110  
 MAIN0111  
 MAIN0115  
 MAIN0116  
 MAIN0117  
 MAIN0118  
 MAIN0120  
 MAIN0130  
 MAIN0142  
 MAIN0150  
 MAIN0153  
 MAIN0155  
 MAIN0156  
 MAIN0158  
 MAIN0160  
 MAIN0170  
 MAIN0180  
 MAIN0190  
 MAIN0200  
 MAIN0210

NMAT0010

```

C      II: THE SUBSCRIPT OF THE COLUMN WITH LEAST NUMBER OF ZEROS
C      NZ: THE NUMBER OF ZEROS IN EACH COLUMN
C      NZCON: COUNTS THE NUMBER OF ZEROS IN A COLUMN THAT ARE NOT DEFINED IN THE
C      BORDER EQUATIONS
C
C      DIMENSION MT(N,NN),IM(N,NN),IT(N),NZ(NN),ISV(NN),NZCON(NN)
C      ZERO THE VALUE OF THE NUMBER OF ZEROS IN EACH COLUMN
C      DO 320 I=1,NN
C      NZCON(I)=0
C      320 NZ(I)=0
C      ADD THE DESIGNATED COLUMN TO THE INITIAL MATRIX AND COUNT THE NUMBER
C      OF ZEROS IN EACH COLUMN
C      DO 300 J=1,NN
C      DO 305 K=1,N
C      MT(K,J)=IM(K,J)+IT(K)
C      IF(MT(K,J).EQ.0) CALL KONFLT(NN,J,K,NZ,NZCON,IM,N)
C      305 CONTINUE
C      IF ANY COLUMN HAS NO ZEROS, A SOLUTION HAS BEEN REACHED
C      IF(NZ(J).EQ.0) GO TO 350
C      IF VARIABLE IS AT MAXIMUM VALUE, DEFINE NUMBER OF ZEROS TO BE AT MAXIMUM SO
C      THAT COLUMN WILL NOT BE FURTHER CONSIDERED
C      IF(ISV(J).EQ.N) NZ(J)=N
C      300 CONTINUE
C      FIND COLUMN WITH LEAST NUMBER OF ZEROS
C      CALL IVAR(NN,NZ,II,NZCCN,N)
C      RETURN
C      A SOLUTION HAS BEEN REACHED
C      350 D=NN
C      II=J
C      WRITE(6,390) II
C      390 FORMAT(/'10X,'LAST VARIABLE USED IS',I35,I3)
C      RETURN
C      END
C
C      SUBROUTINE IVAR(NN,MZ,II,NC,N)
C      SUBROUTINE TO DETERMINE WHICH COLUMN HAS THE LEAST NUMBER OF ZEROS THAT
C      ARE NOT ASSOCIATED WITH BORDER AREAS.
C      SYMBOLS ARE:
C      MZ=NZ; NC=NZCON
C      NP: DUMMY TO INSURE ALL COLUMNS ARE CHECKED
C      II: DENOTES THE COLUMN(VARIABLE) TO BE USED.
C
C      DIMENSION MZ(NN),NC(NN)
C      ZERO VALUES
C      II=1
C      NP=NN-1
C      IF AN AREA IS AT ITS MAXIMUM COLOR, INSURE IT IS NOT CHOSEN

```

NMAT0020  
 NMAT0023  
 NMAT0024  
 NMAT0025  
 OF  
 NMAT0030  
 NMAT0040  
 NMAT0050  
 NMAT0060  
 NMAT0065  
 NMAT0080  
 MAXIMUM SO  
 NMAT0085  
 NMAT0090  
 NMAT0100  
 NMAT0130  
 NMAT0140  
 NMAT0142  
 NMAT0144  
 NMAT0145  
 NMAT0150  
 NMAT0160

IVAR0020  
 IVAR0030  
 IVAR0035



```

C      DO 210 I=1,NN
C      210 IF(MZ(I).EQ.N) NC(I)=N
C      PICK THE COLUMN WITH THE LEAST NUMBER OF NONBORDERING ZERO AREAS
C      DO 200 K=1,NP
C      200 IF(NC(K+1).LT.NC(II)) II=K+1
C      RETURN
C      END

```

```

IVAR00037
IVAR00038
IVAR00040
IVAR00050
IVAR00060
IVAR00070

```

```

C      SUBROUTINE NZERO(N,NN,II,IT,ITSV,MV,E,KL,IA,BA,KM)
C      A SUBPROGRAM TO DETERMINE WHICH EQUATIONS CORRESPOND TO THE VARIABLE
C      CHOSEN. // SYMBOLS ARE AS DEFINED IN MAIN PROGRAM; OTHER SYMBOLS ARE:
C      ITSV: DENOTES VARIABLE IS AT MAXIMUM VALUE
C      BA: DUMMY TO INDICATE THAT 2 ADJACENT AREAS HAVE THE SAME COLOR
C      K2: COUNTER TO INDICATE WHEN ALL BORDER AREAS HAVE BEEN EXAMINED
C      E: A DUMMY TO INDICATE THAT TWO VARIABLES ARE EQUAL AND AT MAX
C      OTHER SYMBOLS: IT=ITAB; KL=KLR
C
C      NZERO0010

```

```

C      DIMENSION IT(N,NN),ITSV(NN),MV(NN),IA(N),KM(NN)
C      ZERO THE COUNTERS AND DUMMIES
C      BA=0.0
C      E=0.0
C      K2=0
C      IF COLUMN CHOSEN IS AT MAXIMUM VALUE, RETURN AND CHOOSE A NEW COLUMN
C      IF(MV(II).EQ.KL) GO TO 610
C      INCREASE VALUE OF VARIABLE BY ONE
C      MV(II)=MV(II)+1
C      IF COLOR IS AT THE MAXIMUM VALUE, MAKE A NOTE OF THE VARIABLE
C      IF(MV(II).EQ.KL) ITSV(II)=N
C      CHECK EACH ROW FOR VARIABLE USED TO CHECK THE COLOR OF THE ADJOINING
C      DO 600 JJ=1,N
C      IF(IT(JJ,II).NE.0) CALL INFEAS(N,NN,MV,IT,II,JJ,KL,IA,BA,KM,K2)
C      IF(BA.NE.0.0) RETURN
C      600 CONTINUE
C      RETURN
C      610 E=N
C      END
C
C      NZERO0020
C      NZERO0024
C      NZERO0025
C      NZERO0026
C      NZERO0035
C      NZERO0036
C      NZERO0037
C      AREAS
C      NZERO0038
C      NZERO0040
C      NZERO0045
C      NZERO0050
C      NZERO0060
C      NZERO0064
C      NZERO0065
C      NZERO0070

```

```

C      SUBROUTINE IBACK(N,NN,IA,KNT,II,NZ,IN,ITSV,D,MV,MT,E,KL,KM,NC,N3,
C      INX3)
C      SUBPROGRAM TO DO THE FOLLOWING:
C      (1) SAVE THE COLUMN ADDED TO THE INITIAL MATRIX
C      (2) SAVE THE INDEX VALUE OF THE COLUMN ADDED TO THE INITIAL MATRIX AND SAVE
C      THE INDICATOR FROM WHERE THE COLUMN CAME (ALPHA OR BETA) AND SAVE THE
C      COLOR OF THE VARIABLE
C      (3) REINDEX THE COLUMN, IF NO OTHER COLUMN CAN BE CHOSEN
C
C      IBAK00010
C      IBAK00011

```





```

C RETURN ONE ITERATION
C 410 KNT=KNT-1
C IF ALL ITERATIONS HAVE BEEN EXAMINED, NO SOLUTION EXISTS
C IF (KNT.EQ.0) GO TO 450
C IF THE VARIABLE WAS AT MAXIMUM VALUE, REZERO THE INDICATOR
C IF (MV(I1).EQ.KL) ITSU(I1)=0
C ZERO THE COUNTERS AND INDICATORS
C L4=0
C L2=0
C KN1=0
C DETERMINE THE COLUMNS ASSOCIATED WITH THE ZERO VALUES IN IADD
C DO 482 I=1,N
C 482 IF (IA(I).EQ.0) CALL KNTVAR(N,NN,IN,I,IITMP,L2)
C IF THE COLOR OF THE ADJACENT AREAS THAT ARE PRESENTLY AT A ZERO COLOR LEVEL
C WERE INCREASED TO THE MAXIMUM COLOR, DETERMINE WHAT AREAS WOULD THEN HAVE
C MATCHING COLORS
C DO 460 J=1,L2
C ASSIGN A TEMPORARY VALUE TO VARIABLE THAT IS AT ZERO LEVEL
C IIS=IITMP(J)
C ASSIGN AN INCREASING COLOR VALUE TO ZERO LEVEL VARIABLES
C DO 461 I=1,KL
C MV(IIS)=I
C ASSIGN INITIAL VALUE TO AN INDICATOR TO SHOW WHEN A COLOR ASSIGNED TO AN
C AREA IS NOT THE SAME AS SOME BORDERING AREA
C K2=KN1
C CHECK EACH ROW FOR THE COLOR USED TO CHECK THE COLOR OF THE ADJOINING
C AREAS.
C DO 462 JJ=1,N
C 462 IF (IN(JJ,IIS).NE.0) CALL LKINF(N,NN,IIS,KN1,IN,MV,MVMCH,JJ)
C IF PRESENT COLOR DOES NOT MATCH WITH ANY BORDER AREAS, STOP THE EXAMINING
C PROCESS AND RETURN TO MAIN PROGRAM.
C IF (KN2.EQ.KN1) CALL NOMCH(N,NN,IIS,II,I,MV,IA,IN,NU,KNT,KL,ITSV,
C 1NX3)
C IF (KN2.EQ.KN1) GO TO 441
C 461 CONTINUE
C REZERO VARIABLE EXAMINED
C MV(IIS)=0
C 460 CONTINUE
C DETERMINE FROM WHICH ITERATION EACH OF THE VARIABLES PICKED ABOVE CAME
C DO 471 J=1,KN1
C ITER(J)=0
C DO 471 I=1,KNT
C IF (MVMCH(J).EQ.KOLSV(N+1,I)) ITER(J)=I
C 471 CONTINUE
C ORDER THE ITERATIONS
C CALL IORDER(N,KN1,ITER)

```

IBAK0100

IBAK0110

IBAK0090

IBAK0091

IBAK0092

IBAK0093

IBAK0094

IBAK0095

IBAK0096

IBAK0097

IBAK0098

IBAK0099

IBAK0100

IBAK0101

IBAK0102

IBAK0103

IBAK0103

IBAK0104

IBAK0105

IBAK0106

IBAK0107

IBAK0108

IBAK0109

IBAK0110

IBAK0111

IBAK0112

IBAK0113

```

C RETURN TO THE MOST RECENT ITERATION DEFINED BY IORDER
411 L4=L4+1
C KNT=ITER(L4)
C IF ALL ITERATIONS HAVE BEEN EXAMINED NO SOLUTION EXISTS
  IF(KNT.LE.C) GO TO 450
AD=0.0
C ASSIGN TO THE VARIABLES THE VALUES THAT WERE ASSIGNED IN ITERATIONS
  C TO THE CHOSEN ITERATION
  LNT=KNT-1
  DO 475 J=1,NN
    ITSV(J)=0
    MV(J)=0
    DO 476 I=1,LNT
      IF(KOLSV(N+1,I).EQ.J) MV(J)=KOLSV(N+3,I)
      IF(MV(J).EQ.KL) ITSV(J)=N
C RETRIEVE THE VARIABLE USED IN CHOSEN ITERATION.
      II=KOLSV(N+1,KNT)
      IF(KOLSV(N+1,KNT).EQ.NN) GO TO 411
C RECONSTRUCT CHOSEN ITERATION MATRIX(MAT)
      DO 425 I=1,NN
        NC(I)=0
        NZ(I)=0
      DO 420 J=1,NN
        DO 419 K=1,N
          MT(K,J)=IN(K,J)+KOLSV(K,LNT)
          IF(MT(K,J).EQ.0) CALL KONFLT(NN,J,K,NZ,NC,IN,N)
419 CONTINUE
420 CONTINUE
C USE NEXT COLUMN WITH SAME NUMBER OF ZEROS, IF NO SUCH COLUMN EXISTS RETURN
  C TO NEXT FEASIBLE ITERATION
  IND=KOLSV(N+2,KNT)
  CALL KVAR(NN,NZ,II,AD,MV,ITSV,NC,IND,NU,KNT,NX3)
  KOLSV(N+2,KNT)=IND
  IF(AD.NE.0.0) GO TO 411

C DETERMINE THE NEW VALUE OF THE VARIABLE, INDICATE IF SOLUTION IS
C INFEASIBLE SO THAT A NEW PATH CAN BE TAKEN, AND DEFINE THE NEW COLUMN
C TO BE ADDED TO THE INITIAL TABLEAU
430 CALL NZERO(N,NN,II,IN,ITSV,MV,E,KL,IA,BA,KM)
  IF(BA.NE.0.0) GO TO 430
C IF COLOR CHOSEN IS AT MAXIMUM VALUE, GO TO ANOTHER COLUMN
  IF(E.NE.0.0) CALL KLRMCH(N,NN,II,AD,MV,NZ,IN,ITSV,IA,E,MT,KL,KOLSV,
1KNT,KM,NC,IND,NU,N3,NX3)
C IF NO OTHER COLUMN CAN BE USED, RETURN TO NEXT FEASIBLE ITERATION
  IF(AD.NE.0.0) GO TO 411
C SAVE NECESSARY VALUES TO BE USED IN RECONSTRUCTING THIS ITERATION IF
441 DO 440 I=1,N

```

IBAK0114  
IBAK0115

IBAK0116  
IBAK0117  
PREVIOUS

IBAK0118  
IBAK0120  
IBAK0121  
IBAK0125  
IBAK0130  
IBAK0132  
IBAK0133

IBAK0157  
IBAK0158

IBAK0160  
IBAK0165  
IBAK0170  
IBAK0180  
IBAK0190  
IBAK0210  
IBAK0220  
IBAK0224  
IBAK0228  
IBAK0230  
RETURN

IBAK0235  
IBAK0240

IBAK0250

IBAK0251  
IBAK0252

IBAK0253  
IBAK0254

IBAK0255  
REQUIRED  
IBAK0258

```

440 KOLSV(I,KNT)=IA(I)
    KOLSV(N+1,KNT)=II
    KOLSV(N+3,KNT)=MV(II)
    RETURN
450 D=NN
    WRITE(6,451)
451 FORMAT(10X,'ALL COMBINATIONS HAVE BEEN TRIED, NO SOLUTION EXISTS.')
```

IBAK0280  
IBAK0290  
IBAK0305  
IBAK0310  
IBAK0320  
IBAK0330  
IBAK0340  
IBAK0350

```

    END
    SUBROUTINE KVAR(NN,MZ,II,AD,MV,ISV,NC,IND,NU,KNT,NX3)
    SUBPROGRAM TO FIND THE NEXT COLUMN IN LINE WITH THE LEAST
    C SYMBOLS ARE: MZ=NZ ; OTHERS AS PREVIOUSLY DEFINED
    C
    C DIMENSION MZ(NN),MV(NN),ISV(NN),NC(NN),NU(NX3)
    C ASSIGN INITIAL VALUES
    ISV(II)=0
    IF(NU(KNT).NE.0) GO TO 501
    NP=NN-1
    C IF ALL COLUMNS WITH NONBORDERING ZEROS HAVE BEEN EXAMINED, EXAMINE TOTAL
    C NUMBER OF ZEROS IN EACH COLUMN FOR NEXT COLUMN TO USE.
    IF(IND.NE.0) GO TO 540
    C CHECK THE REST OF THE COLUMNS IN THE MATRIX FOR THE NEXT COLUMN WITH
    C NUMBER OF NONBORDERING ZEROS.
    DO 520 K=II,NP
    IF(NC(K+1).EQ.NC(II)) GO TO 530
    520 CONTINUE
    C IF NO OTHER NONBORDERING ZERO COLUMN CAN BE CHOSEN, ASSIGN POSITIVE VALUE
    C TO INDICATOR
    IND=NN
    II=1
    C DETERMINE WHICH COLUMN HAS THE LEAST TOTAL NUMBER OF ZEROS
    DO 510 K=1,NP
    510 IF(MZ(K+1).LT.MZ(II)) II=K+1
    RETURN
    C CHECK THE REST OF THE COLUMNS IN THE MATRIX FOR THE NEXT COLUMN WITH
    C TOTAL NUMBER OF ZEROS.
    540 DO 500 K=II,NP
    IF(MZ(K+1).EQ.MZ(II)) GO TO 530
    500 CONTINUE
    C IF NO OTHER COLUMN WITH SAME NUMBER OF ZEROS EXISTS, NOTE SAME AND RETURN
    501 AD=NN
    RETURN
    C DEFINE NEW VARIABLE TO BE USED
    530 II=K+1
    RETURN
    END
    KVAR0010
    KVAR0020
    KVAR0030
    KVAR0035
    KVAR0040
    KVAR0048
    KVAR0050
    KVAR0051
    KVAR0052
    KVAR0053
    KVAR0054
    KVAR0057
    KVAR0058
    KVAR0059
    KVAR0060
    KVAR0070
    KVAR0080
    KVAR0090
    KVAR0100
    KVAR0130
```



```

C SUBROUTINE INFEAS(N,NN,MV,IN,II,J,KL,IA,BA,KM,K2) INFS0010
C SUBROUTINE TO DETERMINE WHICH STATES BORDER ON STATE CHOSEN. SYMBOLS ARE
C AS PREVIOUSLY DEFINED
C
C DIMENSION MV(NN),IN(N,NN),IA(N),KM(NN) INFS0020
C INCREASE COUNTER BY ONE; INDICATES WHEN ALL BORDERS HAVE BEEN CHECKED
K2=K2+1 NZER0033
C IF(II.EQ.1) GO TO 710 INFS0040
C EXAMINE ROW FROM STATE#1 TO CHOSEN STATE MINUS 1 TO SEE IF AN ADJACENT
C STATE IS DEFINED IN THIS AREA
IL=II-1
DO 720 KK=1,IL
C IF THERE EXISTS AN ADJACENT AREA, CHECK THE TWO COLORS FOR A MATCH
IF(IN(J,KK).NE.0) CALL ICHK(N,NN,II,IN,MV,KK,KL,IA,BA,KM,K2)
IF(BA.NE.0.0) RETURN
720 CONTINUE
C IF(II.EQ.NN) RETURN
C EXAMINE ROW FROM CHOSEN STATE PLUS 1 TO STATE#NN TO SEE IF AN ADJACENT
C STATE IS DEFINED IN THIS AREA
710 IU=II+1
DO 725 KK=IU,NN
C IF THERE EXISTS AN ADJACENT AREA, CHECK THE TWO COLORS FOR A MATCH
IF(IN(J,KK).NE.0) CALL ICHK(N,NN,II,IN,MV,KK,KL,IA,BA,KM,K2)
IF(BA.NE.0.0) RETURN
725 CONTINUE
RETURN
END
C
C SUBROUTINE ICHK(N,NN,II,IN,MV,K,KL,IA,BA,KM,K2) ICHK0010
C SUBROUTINE COMPARE THE COLORS OF ADJACENT AREAS. IF THE COLORS ARE THE SAME
C THE ROUTINE RETURNS TO FIND ANOTHER COLOR OR VARIABLE IF NECESSARY.
C WHEN THE COLOR HAS BEEN ESTABLISHED, THE NEW COLUMN TO ADD TO THE INITIAL
C TABLEAU IS DEFINED. SYMBOLS ARE AS PREVIOUSLY DEFINED.
C
C DIMENSION MV(NN),IA(N),IN(N,NN),KM(NN) ICHK0020
C
C IF ADJACENT AREAS ARE THE SAME COLOR, RETURN TO STARTING POINT AND REDEFINE
C COLOR OR VARIABLE
IF(MV(II).EQ.MV(K)) GO TO 830 ICHK0040
C WHEN ALL OF AN AREA'S BORDERS HAVE BEEN CHECKED, DEFINE THE NEW COLUMN TO
C BE ADDED
IF(KM(II).EQ.K2) GO TO 813 ICHK0056
C IF ALL BORDER AREAS HAVE NOT BEEN CHECKED, RETURN TO SUBROUTINE INFEAS
C AND CHECK NEXT BORDER AREA
RETURN ICHK0057

```

```

C      813 DO 820 J=1,N
      820 IA(J)=0
      DO 825 J=1,N
      DO 825 JJ=1,NN
      IA(J)=IA(J)+IN(J,JJ)*MV(JJ)
      825 CONTINUE
      RETURN
      830 BA=N
      RETURN
      END

      SUBROUTINE KLRMCH(N,NN,II,AD,MV,NZ,IN,ITSV,IA,E,MT,KL,KSV,KT,KM,
      INC,IND,NU,N3,NX3)
      SUBROUTINE TO DEFINE ANOTHER COLUMN TO USE, IF THE SOLUTION IS INFEASIBLE
      AND AT MAXIMUM COLOR VALUE. SYMBOLS ARE: AS DEFINED ELSEWHERE

      DIMENSION MV(NN),NZ(NN),IN(N,NN),ITSV(NN),IA(N),MT(N,NN),KM(NN)
      DIMENSION NC(NN),KSV(N3,NX3),NU(NX3)

      C      ZERO DUMMIES USED
      700 E=0.0
      BA=0.0
      KE=KT-1

      C      DEFINE COLOR OF VARIABLE TO BE A PREVIOUS COLOR USED.
      MV(II)=0
      DO 740 I=1,KE
      IF(KSV(N+1,I)).EQ.II) MV(II)=KSV(N+3,I)
      740 USE THE NEXT COLUMN WITH SAME NUMBER OF ZEROS. IF NO SUCH COLUMN EXISTS,
      RETURN TO NEXT APPLICABLE ITERATION
      CALL KVAR(NN,NZ,II,AD,MV,ITSV,NC,IND,NU,KT,NX3)
      C      SAVE THE VALUE OF THE INDICATOR
      KSV(N+2,KT)=IND
      IF(AD.NE.0.0) RETURN OF THE VARIABLE, INDICATE IF SOLUTION IS
      C      DETERMINE THE NEW VALUE OF THE VARIABLE, INDICATE IF SOLUTION IS
      C      INFEASIBLE SO THAT A NEW PATH CAN BE TAKEN, AND DEFINE THE NEW COLUMN
      C      TO BE ADDED TO THE INITIAL TABLEAU
      710 CALL NZERO(N,NN,II,IN,ITSV,MV,E,KL,IA,BA,KM)
      IF(BA.NE.0.0) GO TO 710
      IF(E.NE.0.0) GO TO 700
      C      REZERO PRESENT INDICATOR
      IND=0
      RETURN
      END

      SUBROUTINE KONFLT(NN,J1,K1,NZ,NZCON,IN,N)

```

ICHK00068  
 ICHK00070  
 ICHK00080  
 ICHK00090  
 ICHK00100  
 ICHK00110  
 ICHK00120  
 ICHK00130  
 ICHK00140

KLMH00010  
 KLMH00011  
 KLMH00012  
 KLMH00013  
 KLMH00014

KLMH00020  
 KLMH00021

KLMH00030  
 KLMH00031  
 KLMH00040

KLMH00050  
 KLMH00051  
 KLMH00052

KLMH00060  
 KLMH00065  
 KLMH00070

KLMH00080  
 KLMH00085  
 KLMH00090

KLMH00100  
 KLMH00120

KNFT00010

```

C SUBPROGRAM TO COUNT THE NUMBER OF ZEROS IN EACH COLUMN AND TO COUNT THE
C NUMBER OF ZEROS ASSOCIATED WITH NONBORDERING AREAS. SYMBOLS ARE AS
C PREVIOUSLY DEFINED
C
  DIMENSION NZ(NN),NZCON(NN),IN(N,NN)
  NZ(J1)=NZ(J1)+1
  IF(IN(K1,J1).EQ.0) NZCON(J1)=NZCON(J1)+1
  RETURN
  END
  KNFT0020
  KNFT0030
  KNFT0040
  KNFT0050

C SUBROUTINE NOMCH(N,NN,IIS,II,KK,MV,IA,IN,NU,KNT,KL,ITSV,NX3)
C IN THE EVENT THAT THE COLOR CHOSEN WHEN ENUMERATING ALL POSSIBLE COLORS
C FOR THE CHOSEN AREA DOES NOT MATCH ANY OF THE BORDERING AREAS, THIS
C SUBPROGRAM DEFINES THE VARIABLE, ITS COLOR, AND THE NEW COLUMN TO ADD
C
  DIMENSION MV(NN),IA(N),IN(N,NN),ITSV(NN),NU(NX3)
  C DEFINE CHCSEN VARIABLE
  II=IIS
  C DEFINE VARIABLE'S COLOR
  MV(II)=KK
  C IF COLOR IS AT THE MAXIMUM VALUE, MAKE A NOTE OF THE VARIABLE
  IF(MV(II).EQ.KL) ITSV(II)=N
  C DEFINE THE COLUMN TO ADD
  DO 1200 J=1,N
  1200 IA(J)=0
  DO 1251 J=1,N
  1251 JJ=1,NN
  DO 1250 IA(J)=IA(J)+IN(J,JJ)*MV(JJ)
  1251 CONTINUE
  KNT=KNT+1
  NU(KNT)=N
  RETURN
  END
  NMCH0020
  NMCH0030
  NMCH0040
  NMCH0050
  NMCH0060
  NMCH0070
  NMCH0080
  NMCH0090
  NMCH0100
  NMCH0110
  NMCH0115
  NMCH0118
  NMCH0120

C SUBROUTINE IORDER(N,KN1,ITER)
C SUBPROGRAM TO ORDER THE ITERATIONS CHOSEN FOR BACKTRACKING
C SYMBOLS ARE AS DEFINED IN SUBROUTINE IBAK
C
  DIMENSION ITER(N)
  KN1M1=KN1-1
  DO 1010 J=1,KN1M1
  IG=ITER(J)
  J1=J+1
  DO 1011 I=J1,KN1
  IF(IG.GE.ITER(I)) GO TO 1011
  IG=ITER(I)
  IORD0010
  IORD0020
  IORD0030
  IORD0035
  IORD0036
  IORD0037
  IORD0038
  IORD0039
  IORD0040

```



IORD00041  
 IORD00042  
 IORD00043  
 IORD00070  
 IORD00080

ITER(J)=ITER(J)  
 ITER(J)=IG  
 1011 CONTINUE  
 1010 RETURN  
 END

C SUBROUTINE LKINF(N,NN,IIS,KNI,IN,MV,MVMCH,JJ)  
 C SUBROUTINE TO DETERMINE WHICH STATES BORDER ON CHOSEN STATE. SYMBOLS  
 C ARE AS PREVIOUSLY DEFINED LKIF0010

C DIMENSION IN(N,NN),MV(NN),MVMCH(NN)  
 C IF(IIS.EQ.1) GO TO 910  
 C EXAMINE ROW FROM STATE #1 TO CHOSEN STATE MINUS 1 TO SEE IF AN ADJACENT  
 C STATE IS DEFINED IN THIS AREA.

C ISL=IIS-1  
 C DO 920 KK=1,ISL  
 C IF THERE EXISTS AN ADJACENT AREA, CHECK THE TWO COLORS FOR A MATCH  
 C IF(IN(JJ,KK).NE.0) CALL LKICHK(NN,IIS,KK,KNI,MV,MVMCH)

920 CONTINUE  
 C IF(IIS.EQ.NN) RETURN  
 C EXAMINE ROW FROM CHOSEN STATE PLUS 1 TO STATE #NN TO SEE IF AN ADJACENT  
 C STATE IS DEFINED IN THIS AREA.

910 ISU=IIS+1  
 C DO 925 KK=ISU,NN  
 C IF THERE EXISTS AN ADJACENT AREA, CHECK THE TWO COLORS FOR A MATCH  
 C IF(IN(JJ,KK).NE.0) CALL LKICHK(NN,IIS,KK,KNI,MV,MVMCH)

925 CONTINUE  
 RETURN  
 END

C SUBROUTINE LKICHK(NN,IIS,K,KNI,MV,MVMCH)  
 C SUBROUTINE COMPARES THE COLORS OF ADJACENT AREAS. IF THE COLORS ARE  
 C SAME, THE ROUTINE RECORDS WHICH AREA HAS SAME COLOR AS CHOSEN  
 C AREA. SYMBOLS ARE AS DEFINED IN IBAK LKIK0010

C DIMENSION MV(NN),MVMCH(NN)  
 C COMPARE THE COLORS OF THE ADJACENT AREAS  
 C IF(MV(IIS).EQ.MV(K)) GO TO 890

C RETURN  
 C RECORD THE VARIABLE WITH THE MATCHING COLOR  
 890 KNI=KNI+1  
 C MVMCH(KNI)=K  
 C RETURN  
 C END

SUBROUTINE KNTVAR(N,NN,IN,I,IITMP,L2) KNV00010

LKIK0020  
 LKIK0030  
 LKIK0040  
 LKIK0044  
 LKIK0045  
 LKIK0050

```

C SUBPROGRAM TO DETERMINE WHICH TWO VARIABLES ARE ADJACENT AND STILL HAVE
C COLOR ZERO.  SYMBOLS ARE AS DEFINED IN IBAK
C
      DIMENSION IN(N,NN),IITMP(NN)
      DO 1300 J=1,NN
      IF (IN(I,J).NE.0) L2=L2+1
      1300 IF (IN(I,J).NE.0) IITMP(L2)=J
      RETURN
      END
      KNVR00020
      KNVR00030
      KNVR00040
      KNVR00050
      KNVR00060

```

## BIBLIOGRAPHY

- Courant, R. and Robbins, H., What Is Mathematics?, p. 246-248, Oxford University Press, 1941.
- Dantzig, G. B., "On Significance of Solving L. P. Problem with Some Integer Variables," Econometrica, V. 28, p. 30-44, January 1960.
- Greenberg, H., "An Algorithm for the Computation of Knapsack Function," Journal of Mathematical Analysis and Applications, v. 26, No. 1, p. 159-162, April 1969.
- Greenberg, H., "A Dynamic Programming Solution to the Integer Linear Program," Journal of Mathematical Analysis and Applications, v. 26, No. 2, p. 454-459, May 1969.
- Oystein, O., The Four-Color Problem, Academic Press, 1967.
- Waller, B. E., III, The Color Problem, M. A. Thesis, Vanderbilt University, Nashville, 1965.

# INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2. Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3. Chief of Naval Operations OP-96 Department of the Navy Washington, D. C. 20350	1
4. Asst. Professor H. Greenberg, Code 55Gd (thesis advisor) Department of Operations Analysis Naval Postgraduate School Monterey, California 93940	1
5. LCDR Dennis S. Read, USN (student) USS STERETT (DLG 31) c/o F.P.O. San Francisco, Calif. 96601	1



UNCLASSIFIED

Security Classification

## DOCUMENT CONTROL DATA - R &amp; D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940		2a. REPORT SECURITY CLASSIFICATION Unclassified	
		2b. GROUP	
3. REPORT TITLE An Iterative Process to Solve the Graph-Coloring Problem			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Master's Thesis, October 1969			
5. AUTHOR(S) (First name, middle initial, last name) Dennis S. Read			
6. REPORT DATE October 1969		7a. TOTAL NO. OF PAGES 39	7b. NO. OF REFS 6
8a. CONTRACT OR GRANT NO.		9a. ORIGINATOR'S REPORT NUMBER(S)	
b. PROJECT NO.			
c.		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
d.			
10. DISTRIBUTION STATEMENT This document has been approved for public release and sale; its distribution is unlimited.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940	
13. ABSTRACT <p>The intent of this paper is to describe a method of coloring a map and to present an algorithm for the solution of this problem. A computer program was developed to provide solutions to the problem of coloring a map which consists of a finite number of areas. This algorithm may also be applied to problems other than map-coloring.</p>			

### KEY WORDS

LINK A

LINK B

LINK C

	ROLE
1.	Chairman
2.	Vice Chairman
3.	Secretary
4.	Treasurer
5.	Member
6.	Member
7.	Member
8.	Member
9.	Member
10.	Member
11.	Member
12.	Member
13.	Member
14.	Member
15.	Member
16.	Member
17.	Member
18.	Member
19.	Member
20.	Member
21.	Member
22.	Member
23.	Member
24.	Member
25.	Member
26.	Member
27.	Member
28.	Member
29.	Member
30.	Member
31.	Member
32.	Member
33.	Member
34.	Member
35.	Member
36.	Member
37.	Member
38.	Member
39.	Member
40.	Member
41.	Member
42.	Member
43.	Member
44.	Member
45.	Member
46.	Member
47.	Member
48.	Member
49.	Member
50.	Member
51.	Member
52.	Member
53.	Member
54.	Member
55.	Member
56.	Member
57.	Member
58.	Member
59.	Member
60.	Member
61.	Member
62.	Member
63.	Member
64.	Member
65.	Member
66.	Member
67.	Member
68.	Member
69.	Member
70.	Member
71.	Member
72.	Member
73.	Member
74.	Member
75.	Member
76.	Member
77.	Member
78.	Member
79.	Member
80.	Member
81.	Member
82.	Member
83.	Member
84.	Member
85.	Member
86.	Member
87.	Member
88.	Member
89.	Member
90.	Member
91.	Member
92.	Member
93.	Member
94.	Member
95.	Member
96.	Member
97.	Member
98.	Member
99.	Member
100.	Member

WT

ROLE	NAME	DATE
1. Chairman	Mr. J. H. ...	...
2. Secretary	...	...
3. Treasurer	...	...
4. ...	...	...
5. ...	...	...
6. ...	...	...
7. ...	...	...
8. ...	...	...
9. ...	...	...
10. ...	...	...
11. ...	...	...
12. ...	...	...
13. ...	...	...
14. ...	...	...
15. ...	...	...
16. ...	...	...
17. ...	...	...
18. ...	...	...
19. ...	...	...
20. ...	...	...
21. ...	...	...
22. ...	...	...
23. ...	...	...
24. ...	...	...
25. ...	...	...
26. ...	...	...
27. ...	...	...
28. ...	...	...
29. ...	...	...
30. ...	...	...
31. ...	...	...
32. ...	...	...
33. ...	...	...
34. ...	...	...
35. ...	...	...
36. ...	...	...
37. ...	...	...
38. ...	...	...
39. ...	...	...
40. ...	...	...
41. ...	...	...
42. ...	...	...
43. ...	...	...
44. ...	...	...
45. ...	...	...
46. ...	...	...
47. ...	...	...
48. ...	...	...
49. ...	...	...
50. ...	...	...
51. ...	...	...
52. ...	...	...
53. ...	...	...
54. ...	...	...
55. ...	...	...
56. ...	...	...
57. ...	...	...
58. ...	...	...
59. ...	...	...
60. ...	...	...
61. ...	...	...
62. ...	...	...
63. ...	...	...
64. ...	...	...
65. ...	...	...
66. ...	...	...
67. ...	...	...
68. ...	...	...
69. ...	...	...
70. ...	...	...
71. ...	...	...
72. ...	...	...
73. ...	...	...
74. ...	...	...
75. ...	...	...
76. ...	...	...
77. ...	...	...
78. ...	...	...
79. ...	...	...
80. ...	...	...
81. ...	...	...
82. ...	...	...
83. ...	...	...
84. ...	...	...
85. ...	...	...
86. ...	...	...
87. ...	...	...
88. ...	...	...
89. ...	...	...
90. ...	...	...
91. ...	...	...
92. ...	...	...
93. ...	...	...
94. ...	...	...
95. ...	...	...
96. ...	...	...
97. ...	...	...
98. ...	...	...
99. ...	...	...
100. ...	...	...

WT

NAME	ROLE
Mr. J. Edgar Hoover	Director
Mr. Clegg	Chief of Bureau
Mr. Glavin	Chief of Bureau
Mr. Ladd	Chief of Bureau
Mr. Nichols	Chief of Bureau
Mr. Rosen	Chief of Bureau
Mr. Tracy	Chief of Bureau
Mr. Carson	Chief of Bureau
Mr. Egan	Chief of Bureau
Mr. Gurnea	Chief of Bureau
Mr. Hendon	Chief of Bureau
Mr. Pennington	Chief of Bureau
Mr. Quinn	Chief of Bureau
Mr. Nease	Chief of Bureau
Mr. Gandy	Chief of Bureau

WT

## Four color problem















thesR25005

An iterative process to solve the graph-



3 2768 001 01270 1

DUDLEY KNOX LIBRARY